(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 November 2007 (01.11.2007)

PCT

(10) International Publication Number
WO 2007/122495 A2

(51) International Patent Classification:
H04L 29/06 (2006.01) *G06F 9/48* (2006.01)
G06F 21/00 (2006.01)

(21) International Application Number: PCT/IB2007/001052

(22) International Filing Date: 23 April 2007 (23.04.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/793,934 21 April 2006 (21.04.2006) US

(71) Applicant *(for all designated States except US)*: **AXALTO SA** [FR/FR]; 6, rue de la Verrerie, F-92190 Meudon (FR).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **LU, HongQuian, Karen** [US/FR]; c/o Axalto SA, Intellectual Property Dpt, 6 rue de la Verrerie, F-92190 Meudon (FR).

(74) Common Representative: **AXALTO SA**; c/o Lukasz WLODARCZYK, Intellectual Property DPT, 6, rue de la Verrerie, F-92190 Meudon (FR).

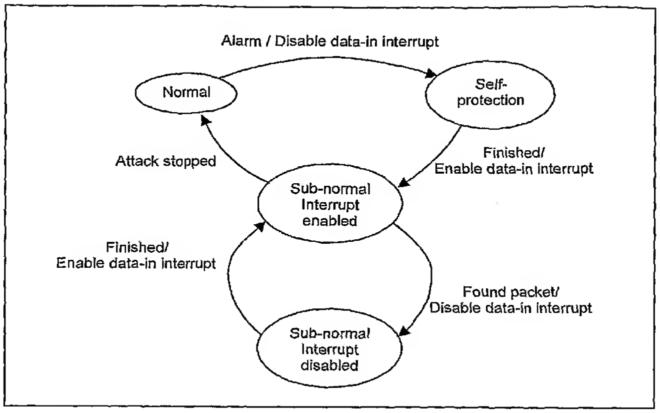
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17:
— *of inventorship (Rule 4.17(iv))*

[Continued on next page]

(54) **Title:** A FRAMEWORK FOR PROTECTING RESOURCE-CONSTRAINED NETWORK DEVICES FROM DENIAL-OF-SERVICE ATTACKS



(57) **Abstract:** The invention relates to a resource constrained network device comprising detection means for detecting DOS attacks, and data-in interrupt means for notifying the device of network data input requests. The device comprises means to disable data-in interrupt means notifications when a DOS attack is detected.

Published:

— *without international search report and to be republished upon receipt of that report*

A Framework for Protecting Resource-Constrained Network Devices from Denial-of-Service Attacks

The invention relates to a resource-constrained network device, and to a method for protecting such resource-constrained network device against DOS attacks (DOS stands for Denial of Service, a well known class of attacks).

Resource-constrained network devices are devices with embedded microprocessors, with very limited computing power and little memory resource, with networking capability, but having relatively low bandwidth. One example of such resource-constrained devices is the network smart card described in PCT/US2004/031572.

A smart card is a plastic card containing an integrated circuit with some memory and a microprocessor. The physical size of a smart card chip is limited to 25 mm2. The first generation of the Network Smart Card, demonstrated by Axalto, Inc. in year 2003, had 6K bytes of RAM and 512K bytes of flash memory. The speed of the microprocessor was 3.75 mHz. The smart card had a standard ISO 7816 interface, which is half-duplex, for communication. Recently, chip manufacturers have been adding full speed USB to smart card chips and have been making higher speed CPUs. For example, an upcoming smart card chip by ST Microelectronics has a full speed USB interface for communication, an expected 33 mHz CPU, 16K bytes of RAM and 64K bytes of EEPROM. Comparing to the state of the art computers, available resources for these small devices are still very limited.

These small devices join the Internet to provide services or to access resources. At the same time, they are exposed to network security threats just as other computers on the network. Because of the limited computing resources and lower network bandwidth, resource-constrained network devices are often more vulnerable to network attacks than other Internet nodes.

One of the network security threats is denial-of-service attack (DoS). Such attacks prevent legitimate users from accessing network resources, such as services on the Internet.

DoS attacks are described in details in particular in:

- Zwicky, E.D., Cooper, S. and Chapman D.B., Building Internet Firewalls, O'Reilly, 2000,
- Chang, R.K.C., "Defending Against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial," IEEE Communication Magazine, October 2002,
- "Denial of Service Attacks," CERT Coordination Center, http://www.cert.org/tech_tips/denial_of_service.html,
- Houle, K.J. and Weaver, G.M. "Trends in Denial of Service Attack Technology," CERT Coordination Center, October 2001, and
- "Distributed Denial of Service (DDoS) Attacks/tools," http://staff.washington.edu/dittrich/misc/ddos/.

The purpose of DoS attacks is to prevent or impair legitimate use of computer or network resources; for example, preventing users from access a popular Internet server. Common targets of resources are bandwidth, processing power, and storage capabilities. The most common DoS attack type is packet-flooding attack, which involves sending a large number of packets to a destination to cause excessive consumptions of resources.

In a distributed denial-of-service (DDoS) attack, a large number of compromised hosts are organized to send packets to a victim computer to consume excessively its resource and/or its Internet connection. There are two kinds of DDoS attacks: direct attacks and reflector attacks. In a direct attack, an attacker arranges many compromised hosts to send a large number of packets directly toward a victim. In a reflector attack, the attacker uses intermediary nodes (routers and servers), called reflectors, to launch the attack. The attacker arranges many compromised hosts to send packets that require responses to reflectors such that the packets' source addresses are set to the victim's IP address. Without realizing the plot, reflectors send responses to the victim consuming the victim's resource.

Common packet types used for DoS or DDoS flooding attacks include the following:

- TCP packets – A flood of TCP packets with various flags set are sent to the victim. Common flags include SYN, ACK, and RST.
- ICMP echo request/reply (also called Ping floods) – A flood of ICMP packets (echo request or echo reply) are sent to the victim.
- UDP packets – A flood of UDP packets are sent to the victim.

The DoS or DDoS attacks can happen at the application layer as well as at network protocol layers. The DoS/DDoS defense mechanism must be build at both application and network protocol layers. For simplicity of the following discussion, we use the word DoS to represent DoS and DDoS.

Large-scale DoS attacks could potentially paralyze the Internet.

Examples of flooding based attacks comprise SYN flooding and ICMP flooding. Various mechanisms are developed for prevention, detection and response to DoS attacks. This is explained in particular in:

- "Denial of Service Attacks," CERT Coordination Center, http://www.cert.org/tech_tips/denial_of_service.html,
- Houle, K.J. and Weaver, G.M. "Trends in Denial of Service Attack Technology," CERT Coordination Center, October 2001, and
- "Distributed Denial of Service (DDoS) Attacks/tools," http://staff.washington.edu/dittrich/misc/ddos/.

These mechanisms together with other security means, such as firewall and intrusion detection system, provide managed security services to enterprises or ISP networks, which consists of routers, servers, and hosts.

Much research work have been done to defend or to mitigate DoS or DDoS attacks in the last decade as explained in particular in

- Chang, R.K.C., "Defending Against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial," IEEE Communication Magazine, October 2002,
- "Denial of Service Attacks," CERT Coordination Center, http://www.cert.org/tech_tips/denial_of_service.html,
- Houle, K.J. and Weaver, G.M. "Trends in Denial of Service Attack Technology," CERT Coordination Center, October 2001,

- Karig, D. and Lee, R., "Remote Denial of Service Attacks and Countermeasures," Princeton University Department of Electrical Engineering Technical Report CE-L2001-002, October 2001, http://www.princeton.edu/~rblee/ELE572Papers/karig01DoS.pdf,
- Khattab, S.M., etc. "Proactive Server Roaming for Mitigating Denial-of-Service Attacks," 2003, http://www.cs.pitt.edu/NETSEC/publications_files/itre03.pdf, and
- "Distributed Denial of Service (DDoS) Attacks/tools," http://staff.washington.edu/dittrich/misc/ddos/.

One basic approach is to detect DoS attack and to filter out attack packets. The Internet infrastructure is hierarchical. The attack detection and attack packets filtering can be done at different levels of the network, for example, at local computer, local network, local ISP network, or upstream ISP network. The effectiveness of the attack detection and packet filtering is dependent on the network level or levels that the defensive mechanisms are executed.

Existing research mainly focuses on protecting enterprise or ISP networks that consist of servers, routers, and host computers. Most approaches are complex; some involve changing Internet infrastructure routers and others require changing servers and clients. Very little literature is found to protect resource constrained embedded network devices from DoS attacks. Lee etc. proposed a port hopping method that can be used for embedded devices, as explained in Lee, H.C.J. and Thing, V.L.L., "Port Hopping for Resilient Networks," 2004, http://www.diadem-firewall.org/publications/VTC2004Fall_Port-Hopping.pdf. With the port hopping, the UDP/TCP port number used by the server varies as a function of time and a shared secret between the server and the client. The method simplifies the detection and filtering of malicious packets. It should work if a network device has agreed with its client or server on the port hopping mechanism. However, because this method requires both clients and their server implement the port hopping, it does not work if a network device provides a standard service, such as a web server, and allows access from a standard client; or if a network device is a standard client, such as web browser, and is to access a standard server.

Many commercial security products, such as intrusion detection system (IDS) are available that can be placed in networks to secure and protect the networks. Internet Firewalls, global defense infrastructures, are used to protect Internet. These approaches protect the embedded network devices in the way that they protect the network. However, the embedded network devices are still in danger if the devices are connected to the Internet via host computers. A host computer may launch, knowingly or unknowingly, DoS attacks against connecting devices. For example, a maliciously installed malware or a computer worm on a host computer may launch such attacks. Because attack packets may not go to the outer network, defense mechanisms setup at the network level or at routers will not be able to help. The resource constrained network devices much have their own defense mechanisms that can live with the limitations of the devices.

Existing designs of the mechanisms are typically not suitable or not as effective for small resource constrained network devices because of the limited resources and bandwidth. Even the computer level DoS defending mechanisms may not be effective for embedded systems. For example, the SYN cookies approach developed for Linux avoids memory consumption for half-open connections. This prevents SYN flooding-based DoS attack from the computer memory perspective. However, it still takes CPU power to react to SYN messages. This may work fine for modern computers. For small resource constrained network devices, all CPU power may be used to respond to SYN messages. The applications on the device cannot get the CPU time. The attacker still achieves the goal of DoS attack. On the other hand, if there are no appropriate security measures for resource constrained network devices; users will be reluctant to connect their devices to the Internet due to security concerns.

Two common DoS prevention mechanisms have been devised, and tailored to resource-constrained network devices. Other prevention methods may also be added.

The first one is known as packet filtering, and is a network security method, which, as its name says, filters incoming or outgoing packets to let good packets pass and to block suspicious packets. Filter rules specify what packets to pass and what packets to reject, thus controlling the packet filtering behavior. Packet filtering helps to protect network devices from DoS attacks to a certain extent as it can filter out potential DoS attack packets.

A multi-stage packet filtering method for resource-constrained network devices is described in US 11/246,736. This method is used as a security measure as well as a memory management scheme to deal with the limited memory resource.

The second method is known as SYN Cookies. Currently Linux kernel and three BSD's (Open, Free, Net) include a facility called SYN cookies, which is used to prevent SYN flooding attack and was proposed in Bernstein, D.J., SYN Cookies, http://cr.yp.to/syncookies.html. Zuquete proposed an improvement to SYN cookies along with details of the Linux SYN cookies implementation, as explained in Zuquete, A., “Improving the Functionality of SYN Cookies,” http://www.inesc-id.pt/pt/indicadores/Ficheiros/165.pdf. The improvement is significant if all TCP implementations follow TCP specification. Unfortunately, this is not the case. Therefore the success of the improved SYN cookies depends on TCP client implementations. Nevertheless, SYN cookies are suitable for resource-constrained network devices, especially when the device has hardware or efficient software implementation of a hash function.

One of the elements encoded in a SYN cookie represents the requesting TCP client's maximum segment size (MSS), which specifies the maximum segment size that the client can receive. The data is a 3-bit encoding of 8 predefined MSS values. The client's MSS is approximated by one of these 8 MSS values. The MSS information is encoded in a SYN cookie because the MSS option only appears in a TCP SYN segment. A resource-constrained network device has very limited memory resource. It is unlikely to exceed its TCP client's MSS. Therefore, the MSS encoding in SYN cookies may not be needed. Based on the risk analysis in Zuquete, A., "Improving the Functionality of SYN Cookies," http://www.inesc-id.pt/pt/indicadores/Ficheiros/165.pdf, not including MSS in SYN cookies can reduce the probability of SYN guessing, which is another network-based attack.

The Linux SYN cookie includes client's ISN. However, the client's ISN may be random, which is the case, for example, Firefox web browser. The original reason for including client's ISN is for the cookie to increase at least as fast as the client's ISN. Because it is random, there is no need to include client's ISN.

The SYN cookies facility may be turned on dynamically. In normal operation, SYN cookies are not necessary. The network device may use SYN cookies when there is a suspicion of SYN flooding attack.

Methods have also been devised for network-Based Attack Monitoring and Detection. Although it is possible to make DoS attacks harder to carry out, quite often DoS attackers are still able to find ways to carry them out. Small resource constrained network devices are more vulnerable because they have very limited computing resource and bandwidth. When under attack, the device shall be able to detect DoS attacks.

In order to defend against network-based attacks, many attack detection or intrusion detections methods are developed. Most existing detection methods are based on statistics of the packets or signatures (or patterns) of packets flows as explained in particular in:

- Chang, R.K.C., "Defending Against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial," IEEE Communication Magazine, October 2002,
- A. Habib, M. Hefeeda, and B. Bhargava, "Detecting Service Violations and DoS Attacks," NDSS Conference Proceedings, Internet Society, 2003,
- Karig, D. and Lee, R., "Remote Denial of Service Attacks and Countermeasures," Princeton University Department of

Electrical Engineering Technical Report CE-L2001-002, October 2001. http://www.princeton.edu/~rblee/ELE572Papers/karig01DoS.pdf,

- Porras, P. and Neumann, P. "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," National Information Security Conference, 1997, http://www.sdl.sri.com/projects/emerald/emerald-niss97.html,
- Siaterlis, C. and Maglaris, B., "Towards Multisensor Data Fusion for DoS Detection," The 19th Annual ACM Symposium on Applied Computing, March 2004, http://www.netmode.ntua.gr/papers/papers/siaterlis_sac04.pdf, and
- Valdes, A., Skinner, K., "Adaptive, Model-based Monitoring for Cyber Attack Detection," http://www.sdl.sri.com/papers/a/d/adaptbn/adaptbn.pdf.

These methods require the understanding of normal or abnormal packets probabilities or signatures and being able to separate normal and abnormal packet flows. The systems need to be trained and be adaptable to changes of the statistics or signatures. Most of these methods deal with known attacks well but have difficulties to unknown attacks. Resource-constrained network devices can only adopt some existing methods if they are not resource consuming nor computational expensive.

From the implementation perspective, most existing attack detection methods are implemented as network components. Research has shown that detection upstream in a network is more effective than detection down stream. On the other hand, detection in host computers is also important because network-based attacks may be in the local network. Existing designs of detection methods in host computers typically run as separated processes or separate tasks. This means the detection module runs all the time when it is turned on. A resource-constrained embedded network device cannot afford to use this kind of approach because the detection module would compete with the device's main task for the very limited computing source.

A detection method for network-based attacks, designed for small resource-constrained network devices, but also applicable to larger computer systems, has been proposed in Lu, H.K., "A Method of Detecting Network-based Attacks for Resource-Constrained Network Devices," PCT/IB2006/003650.

It is based on the system's operational behavior, instead of packets probabilities or signatures. The model expresses expected operational behaviors and attack-suspicious behaviors of the system. Real-time monitors raise an alarm when the system has deviated from the expected operations or when attack-suspicious behaviors occur.

This detection mechanism is embedded inside the network module. This provides real-time detection and conserves memory. The small monitoring code only runs when the code execution passes through there. Therefore, no separate task is required for the detection module and the impact on the system performance is minimum. Additional advantages of this approach include the ability to detect unknown attacks, simplicity, extensibility, and flexibility. The effectiveness of the detection does not depend on known packets probabilities or signatures. Furthermore, the method can be combined with other existing packets based approaches.

When a resource-constrained network device is under flooding-based DoS attack, it is busy reacting to data input (data-in) requests. It typically does not have enough CPU to do anything else, and the DoS attack succeeds. There's a known approach of server roaming described in

- Khattab, S.M., etc. "Proactive Server Roaming for Mitigating Denial-of-Service Attacks," 2003, http://www.cs.pitt.edu/NETSEC/publications_files/itre03.pdf, and
- Khattab, S.M., etc. "Roaming Honeypots for Mitigating Service-level Denial-of-Service Attacks," 24th International Conference on Distributed Computing Systems (ICDCS'04).

But this approach turns out not to be helpful because the network device here is most likely a personal device that cannot be replaced without strict procedures.

It is an aim of the invention to provide a framework for resource constrained network device, for further protecting resource constrained network devices from DoS attacks. The invention involves an interrupt-based method for the resource-constrained network devices to respond in particular to flooding-based DoS attacks. The framework according to preferred embodiments of the invention comprises methods for prevention, detection, self-protection, continuation, and mitigation.

According to the invention, the problem of better protecting a resource constrained network devices from DoS attacks is solved by having a resource constrained network device comprise detection means for detecting DOS attacks, and data-in interrupt means for notifying the device of network data input requests, and means to disable data-in interrupt means notifications when a DOS attack is detected. This is advantageous because during the DOS attack, the device is no longer overloaded with input requests, and can find the time to execute vital tasks, such as protection tasks etc.

The invention and its advantages will be explained more in details in the following specification referring to the appended drawings, in which:

Figure 1 represents a framework for protecting resource-constrained network devices from DoS attacks,

Figure 2 represents interactions among the components of the protection framework of Figure 1 and other software modules of a device according to a preferred embodiment of the invention,

Figure 3 represents operational states illustrating how a response mechanism according to the invention may find enough time to work properly by managing data input interrupts,

Figure 4 shows how a network module of a device according to a preferred embodiment of the invention may get some CPU time during a self-protection procedure, Figure 5 shows some options offered to users of devices according to preferred embodiments of the invention.

According to a preferred embodiment of the invention, a framework for protecting resource constrained network devices from DoS attacks, consists of several components for prevention, detection, self-protection, continuation, and mitigation, as illustrated in Figure 1.

This protection framework interacts with software modules of the resource-constrained network device to protect the system, data, applications, and the services on the device. Figure 2 illustrates the interactions among components of the protection framework and other software modules of the device.

The protection module provides measures to prevent DoS attacks, such as packet filtering and TCP Sync cookie. The detection module monitors the whole systems operational behavior to detect possible DoS attacks. The module can also include traditional detection methods by monitoring the statistics or signatures of the packets. Once a possible attack is detected, the detection module informs the operating system, which invokes the self-protection procedure. The network stack may also try to continue legitimate communications to enable applications to continue network activities. The continuation is supported by managing I/O interrupts, active packet filtering and a mitigation method that tries to stop DoS attack.

Self-protection aims to protect the data and the applications on the network device. Continuation aims to continue the service if possible. Mitigation tries to stop the DoS attack.

According to a preferred embodiment, when an attack is detected and the alert is sent to the OS, the OS disables the hardware interrupt for data-in. This gives the system CPU to run the self-protection procedures, which are described below. Once such procedures finish, the OS enables the hardware interrupt for data-in. The corresponding interrupt handler and the network module can function again. Because the device is still under attack, the network module actively filters out unwanted packets as early as possible and actively looks for expected packets for outstanding connections. Once an expected packet arrives, the hardware data-in interrupt is disabled again. The network module processes the packet, and may pass the data to the application. Once the packet is processed and consumed, the hardware data-in interrupt is enabled again, as illustrated on Figure 3. Because the TCP is a reliable transmission protocol and the device can process packet fast enough, missed packets will be limited. Even if packets were missed, they will be retransmitted again from the other end.

According to a preferred embodiment, once the network module of the resource-constrained network device detects a network-based attack, it alerts the operating system. It may voluntarily suspend its task to give the CPU to other applications. The operating system invokes self-protection procedures, which alert and schedule the tasks to protect data, files, and applications. The self-protection procedures include securing sensitive in-memory data to secure storage, finishing outstanding file transactions, saving future-needed application contexts, and ending certain tasks or applications.

As mentioned earlier, the operating system disables the data input interrupts to gain CPU time for the self-protection procedures. The operating system may choose to execute the self-protection procedures uninterrupted by disabling timer interrupts or to allow some interruptions by allowing timer interrupts. Even in the latter case, very limited timer handlers are enabled. In this case, the network module can set up a timer to get some CPU time. The timer interrupt handler temporarily enables the data-in interrupt to enable the network module to catch incoming packets. The network module drops all incoming packets, except the TCP packets with ACK set for previous sent messages. Such ACK packets free the send buffers. They also enable the network module to release CPU to applications tasks. For example, this enables the socket send() function to return back to its call, as illustrated on Figure 4. The network module should only do some very quick thing in this situation and return control to the operating system. The operating system disables the data-in interrupt to continue the self-protection procedures.

Once the self-protection procedures finish, the OS enables the hardware interrupt for data-in. The corresponding interrupt handler and the network module can function again. The network module finishes the queued task as much as it can, for example, send out queued out-going messages. If the network device is a server, it may send a message to its clients, which will be described below.

The network module preferably continues dropping all incoming packets as early as possible, for example, at the interrupt level. The network module continues to filter out unwanted packets, performs its network stack work and enables the applications to work, which is described below.

Using the detection and self-protection methods described earlier, when a resource-constrained network device is under a network-based attack, it is still alive. During and after executing self-protection procedures, the network module is given some CPU time and may finish sending pending messages. The device might not miss much useful packets even though the incoming packets were dropped. The Internet server or client that the device was connected with will resend the messages, if the connections are not timed out yet, because the TCP is a reliable transmission protocol. At this time, there are several options for continuation, as illustrated on Figure 5. The network device can inform the user these options via sending a message, for example, sending a web page to the browser that the user is using.

The message can warn the user that the device is under a DoS attack and provide options and instructions to the user. The user can choose to disconnect the device or to continue to use the device. If he chooses to disconnect the device, he does so, for example, by taking out his network device, and tries again some other time or uses a different computer. If he chooses to continue using the device, he has several options: continue with the current session, start a new session, or finish the current session and then start a new session. There are, at least, two ways to start a new session. The continuation method is described below. The new session method is described further below.

When the network device finished the safety procedure as explained above, the network module of the device can start to process the packets again. If the DoS attack has stopped, the device goes back to the normal operation. If the attack is still going on, the network module drops all incoming packets except those from the Internet client or server that the network device is communicating with. With the detection system's feedback, the front-end filter can filter out offending packets. For example, if SYN flooding attack is detected, all SYN packets may be filtered out, except those expected ones. Practically, this is a control system.

During this continuation, the data-in interrupt is turned on and off as explained earlier, and as illustrated in Figure 4. The device might be slow, but it should continue to function. The warning message to the user preferably informs the user that the operation might be slow. When the current session is finished, the device preferably provides the user an opportunity to start a new session.

How to start a new communication session in order to mitigate the DoS attack is described below. The method may stop the DoS attack.

Resource constrained network devices may be implemented in different ways. In most cases, a device can choose its own IP address. A device might even have more than one IP address. From the host perspective, the device is a network. In these cases, when the network device is under a DoS attack and the user has decided to start a new communication session, the device can enable IP hopping, as explained in particular in Jones, J., "Distributed denial of service attacks: Defenses, a special publication," Global Integrity, Technical Report, 2000. It basically discards its current IP address and assigns itself a new IP address. If the DoS attack is a targeted attack, for example, SYN flooding, the device can simple drop the messages for the old IP address initially. Once the host has learned the new IP address and updated its ARP cache and its routing table, the packets for the old IP address will not come to the device because the device has a different IP address now. This stops the DoS attack. In order for this to work, the invention solves the two following problems:

(1) how to inform the user of the new IP address

(2) how to inform the host of the new IP address

The network device provides a web server and the user interacts with his device through a standard web browser. As mentioned earlier, when under a DoS attack, the device sends a warning and instruction web page to the browser. We then assume that the user has chosen to start a new session. According to the invention, there are at least two ways for the user to start a new session:

(1) The warning and instruction web page contains an http (or https) link that points to the new URL (with the new IP address) of the device. The user can click the link to start a new session.

(2) Some USB network devices, such as the Network Smart Card, allow the user to get to the login web page of the device by click an icon through the corresponding mass storage interface. In this case, the user closes the current browser, goes back to the device's mass storage device interface, and clicks the icon to start a new session.

In both methods, the user does not need to know the new IP address of the network device. The device either sets the link in the warning and instruction web page, in method 1, or changes its address in the startup file that is clicked by the user from the mass storage device interface, in method 2. The browser will use the new address specified in the link or in the startup file to open the device's login page.

During the process, the host will send an ARP asking who has this IP address. The network device will respond. The host will automatically update its ARP cache and routing table to reflect the device's IP address. The device can now function using the new IP address.

The old IP address of the network device will be eventually dropped out from the host's ARP cache and routing table. The flooding packets of the DoS attack will then no longer be sent to the device. This technique thus stops the DoS attack. In our experiments, we used a Windows XP laptop as a host computer to run a web browser to access the web server in the network device. Two or three minutes after the network device changed its IP address, the old IP address was dropped out from the host's ARP cache. This was observed from both "arp –a" shell command and from a protocol analyzer ethereal.

The approach described here has advantages over existing IP hopping, which protect a public server, whose clients use Domain Name Server (DNS) to lookup the server's IP address. Before the DNS updates its cache for the IP address change, clients' messages may be filtered out by a network firewall. The clients cannot reach the server during this period. Therefore, the IP hopping has latencies. For the method proposed here, the user is actively involved in the process to make decisions and to make the transition happen directly. The actual new IP address is used to make a new connection instead of through DNS. In addition, the network device, such as a network smart card, may be clients as well as servers, both of which can take advantage of IP hopping.

CLAIMS

1. A resource constrained network device comprising detection means for detecting DOS attacks, and data-in interrupt means for notifying the device of network data input requests, characterized in that it comprises means to disable data-in interrupts when a DOS attack is detected.
2. The resource constrained network device according to claim 1, comprising self protection means for protecting the contents of said device in case of DOS attacks.
3. The resource constrained network device according to claim 1, comprising continuation means for continuing operation of said device despite the DOS attack.
4. The resource constrained network device according to claim 1, comprising mitigation means set to stop the current communication session and to start a new communication session in order to escape the current DOS attack.
5. The resource constrained network device according to claim 2, wherein said device comprises an operating system, said operating system being set to enable access to data stored in said device, the self protection means being set to notify the operating system of the attack in order to protect said data.
6. The resource constrained network device according to claim 5, wherein said device comprises memory and secure storage, wherein the self protection means comprise means for securing sensitive data located in memory into secure storage.
7. The resource constrained network device according to claim 5, wherein the operating system is set to manage a file system, and wherein the self protection means comprise means for finishing outstanding file transactions.
8. The resource constrained network device according to claim 2, wherein said device comprises an operating system, said operating system being set to enable access to applications stored in said device, the self

protection means being set to notify the operating system of the attack in order to protect said applications.

9. The resource constrained network device according to claim 8, wherein the self protection means comprise means for ending certain tasks or applications.

10. The resource constrained network device according to claim 8, wherein the self protection means comprise means for saving application contexts which may be needed in the future.

11. The resource constrained network device according to claim 2, wherein after self protection of the device against a DOS attack by self protection means, the data-in interrupts are enabled again.

12. The resource constrained network device according to claim 1, comprising timer interrupt means for notifying the device at times defined by a timer, and comprising means to enable timer interrupts when a DOS attack is detected.

13. The resource constrained network device according to claim 12, wherein the device is set to set up the timer in order to allocate only a small fraction of the CPU time for processing network data input requests while a DOS attack is underway.

14. The resource constrained network device according to claim 1 or 13, comprising means to temporarily enable data-in interrupts after a DOS attack is detected in order to enable the processing of some input data.

15. The resource constrained network device according to claim 14, wherein the device uses the TCP/IP protocol, and wherein upon detection of a DOS attack, the device is set to drop all network data input requests, except those corresponding to TCP packets with ACK set for previous sent messages, in order to free the send buffers.

16. The resource constrained network device according to claim 1, wherein upon detection of a DOS attack the device is set to send a message to the user of said device in order to inform him of the DOS attack.

17. The resource constrained network device according to claim 1, wherein said device comprises a server, and wherein upon detection of a DOS attack, the device is set to send a message to clients connected to said server.

18. The resource constrained network device according to claim 16 or 17, wherein the message consists of a web page designed to be displayed to the user of said device in order to inform him of the DOS attack.

19. The resource constrained network device according to claim 3, wherein upon DOS attack detection the continuation means are set to continue with the current session, or to start a new session, or to finish the current session and then start a new session.

20. The resource constrained network device according to claims 16 and 19, wherein the message sent to the user contains a request to select between at least two of the following options:

    a. continuing with the current session, or
    b. starting a new session, or
    c. finishing the current session and then starting a new session,

    and wherein the continuation means are set to proceed according to the selection.

21. The resource constrained network device according to claim 20, wherein the message comprises a link associated with each option, and wherein the selection is done by clicking the relevant link.

22. The resource constrained network device according to claim 4, wherein the device uses the IP protocol, and wherein upon DOS attack detection the mitigation means are set to discard the current IP address of said device and to assign a new IP address.

23. The resource constrained network device according to claims 16 and 22, wherein the message sent to the user contains the new IP address of said device.

24. The resource constrained network device according to claim 23, wherein said new IP address is associated with a link, wherein said link is set to be displayed in the message, and wherein clicking said link triggers a new communication session with said device using the new IP address.

25. The resource constrained network device according to claim 22, wherein said device is a USB device comprising a mass storage interface for accessing said device from a host, and wherein upon connection to said device through the mass storage interface, the mass storage interface is

set to initiate a new communication session with said device using the new IP address.

26. The resource constrained network device according to any previous claim, wherein said device is a smart card.

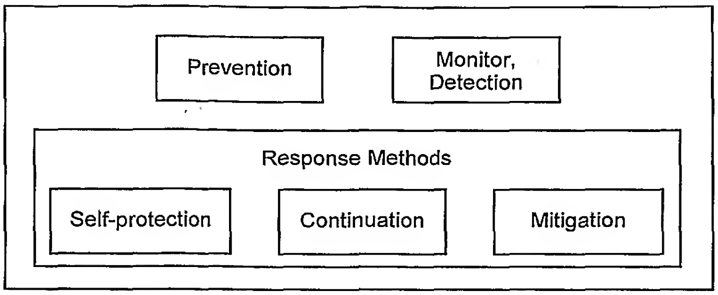


Figure 1

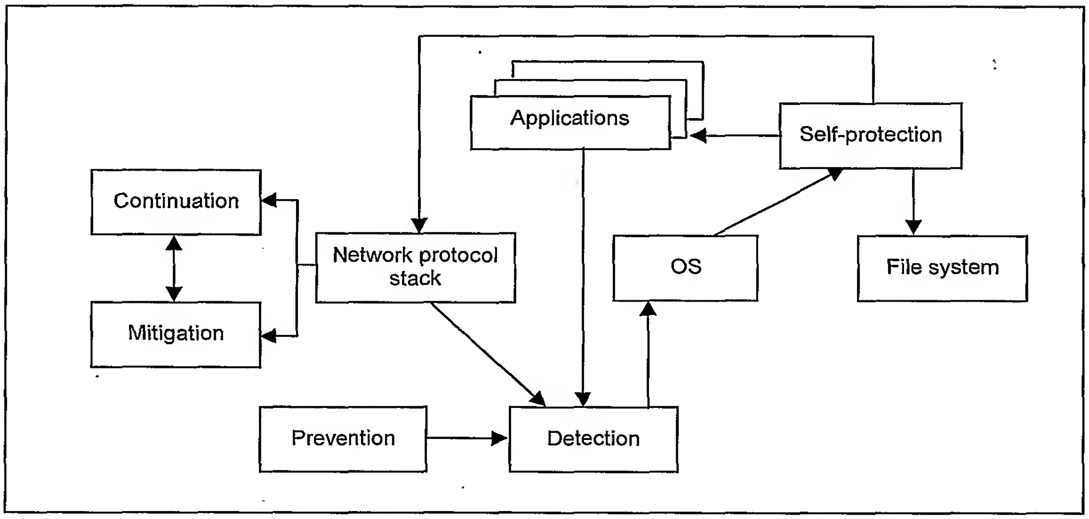


Figure 2

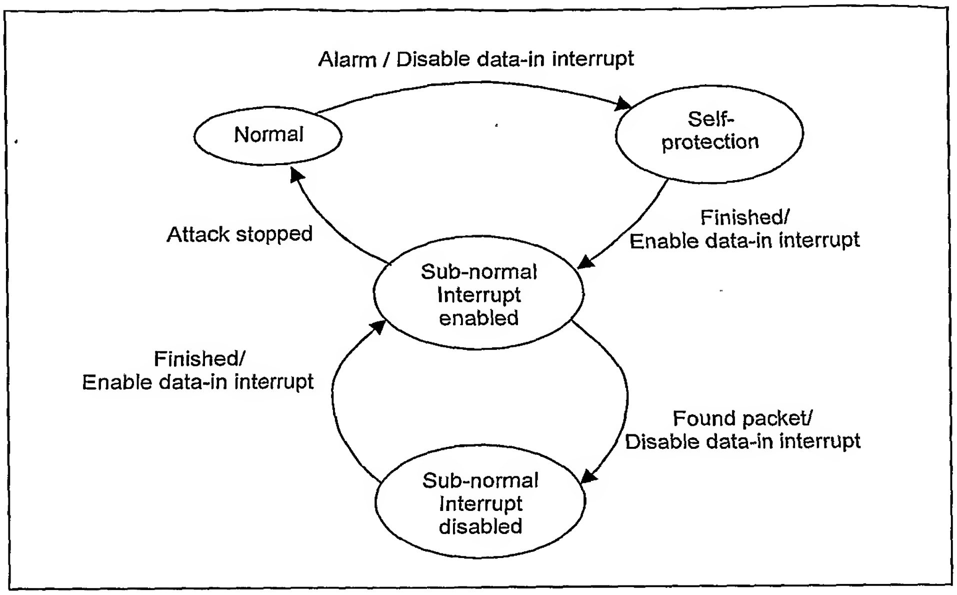

Alarm / Disable data-in interrupt
Normal
Self-protection
Attack stopped
Finished/
Enable data-in interrupt
Sub-normal Interrupt enabled
Finished/
Enable data-in interrupt
Found packet/
Disable data-in interrupt
Sub-normal Interrupt disabled

Figure 3

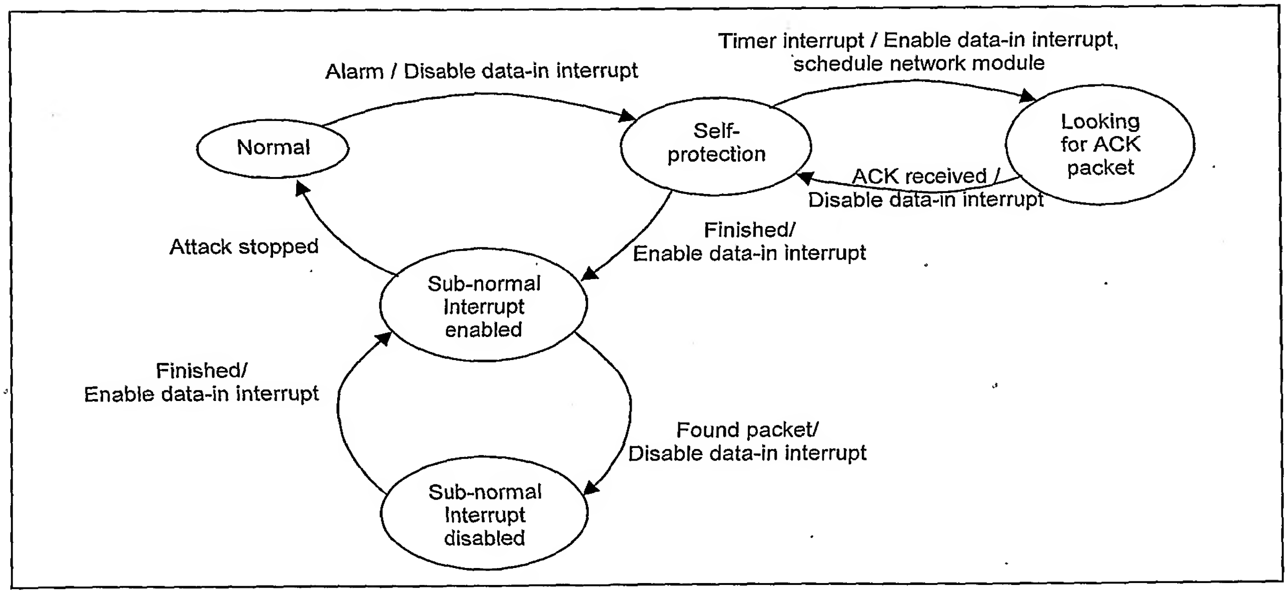

Timer interrupt / Enable data-in interrupt, schedule network module
Alarm / Disable data-in interrupt
Normal
Self-protection
Looking for ACK packet
ACK received /
Disable data-in interrupt
Attack stopped
Finished/
Enable data-in interrupt
Sub-normal Interrupt enabled
Finished/
Enable data-in interrupt
Found packet/
Disable data-in interrupt
Sub-normal Interrupt disabled

Figure 4

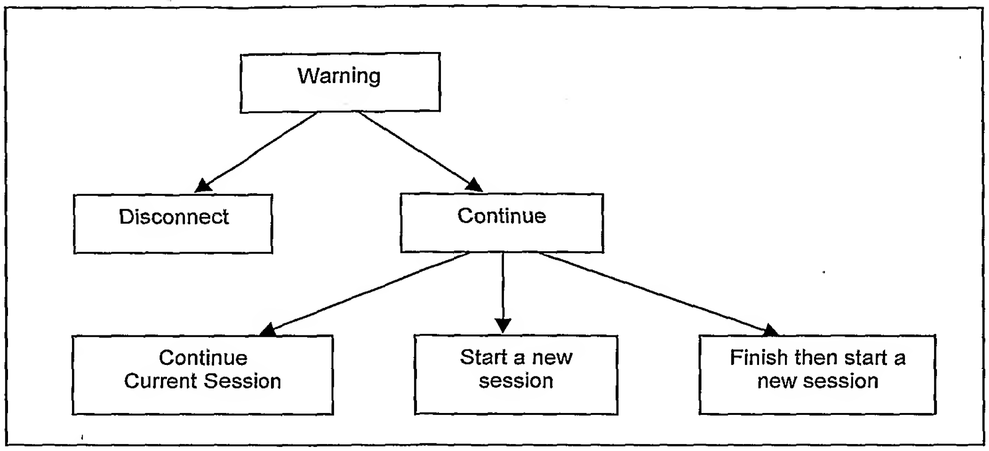

Warning
Disconnect
Continue
Continue Current Session
Start a new session
Finish then start a new session

Figure 5

DERWENT-ACC-NO: 2007-816533

DERWENT-WEEK: 200807

TITLE: Resource constrained network device e.g. network smart card, for e.g. intrusion detection system, has detection module monitoring system`s operational behavior, and operating system set to enable access to data

INVENTOR: LU H K

PATENT-ASSIGNEE: AXALTO SA[AXALN]

PRIORITY-DATA: 2006US-793934P (April 21, 2006)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE
WO 2007122495 A2	November 1, 2007	EN
WO 2007122495 A3	January 17, 2008	EN

DESIGNATED-STATES: AE AG AL AM AT AU AZ BA BB BG BH
BR BW BY BZ CA CH CN CO CR CU CZ
DE DK DM DZ EC EE EG ES FI GB GD GE
GH GM GT HN HR HU ID IL IN IS JP KE
KG KM KN KP KR KZ LA LC LK LR LS
LT LU LY MA MD MG MK MN MW MX
MY MZ NA N G NI NO NZ OM PG PH PL
PT RO RS RU SC SD SE SG SK SL SM SV
SY TJ TM TN TR TT TZ UA UG US UZ VC
VN ZA ZM ZW AT BE BG BW CH CY CZ
DE DK EA EE ES FI FR GB GH GM GR HU
IE IS IT KE LS LT LU LV MC MT MW MZ

NA NL OA PL PT RO SD SE SI SK SL SZ
TR TZ UG ZM ZW AE A G AL AM AT AU
AZ BA BB BG BH BR BW BY BZ CA CH
CN CO CR CU CZ DE DK DM DZ EC EE
EG ES FI GB GD GE GH GM GT HN HR
HU ID IL IN IS JP KE KG KM KN KP KR
KZ LA LC LK LR LS LT LU LY MA MD
MG MK MN MW MX MY MZ NA NG NI
NO NZ OM PG PH PL PT RO RS RU SC SD
SE SG S K SL SM SV SY TJ TM TN TR TT
TZ UA UG US UZ VC VN ZA ZM ZW AT
BE BG BW CH CY CZ DE DK EA EE ES FI
FR GB GH GM GR HU IE IS IT KE LS LT
LU LV MC MT MW MZ NA NL OA PL PT
RO SD SE SI SK SL SZ TR TZ UG ZM ZW

APPLICATION-DATA:

PUB-NO	APPL-DESCRIPTOR	APPL-NO	APPL-DATE
WO2007122495A2	N/A	2007WO-IB001052	April 23, 2007

INT-CL-CURRENT:

TYPE	IPC DATE
CIPP	H04L29/06 20060101
CIPS	G06F21/00 20060101
CIPS	G06F9/48 20060101

ABSTRACTED-PUB-NO: WO 2007122495 A2

BASIC-ABSTRACT:

NOVELTY - The device has a detection module monitoring a system`s operational behavior to detect possible denial of service (DoS) attacks. An operating system is set to enable access to data stored in the device. A network module continues dropping all incoming packets at an interrupt level. The network module continues to filter out unwanted packets. The operating system disables data input interrupts to gain CPU time for self-protection procedures.

USE - Used for a commercial security product such as intrusion detection system.

ADVANTAGE - The operating system protects resource constrained network devices from denial of service (DoS) attacks. The device is not overloaded with input requests, and finds the time to execute vital tasks such as protection tasks.

DESCRIPTION OF DRAWING(S) - The drawing shows an operational state illustrating how a response mechanism finds enough time to work properly by managing data input interrupts.

CHOSEN-DRAWING:	Dwg.3/5
TITLE-TERMS:	RESOURCE CONSTRAIN NETWORK DEVICE SMART CARD INTRUDE DETECT SYSTEM MODULE MONITOR OPERATE BEHAVE SET ENABLE ACCESS DATA

DERWENT-CLASS: T01 T04

EPI-CODES: T01-F02A1; T01-F05G; T01-H07A; T01-J12; T01-N01A2C; T01-N02B1D; T01-N02B1E; T01-N02B2B; T04-K03A;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: 2007-649233